

Improving the performance of eye trackers with limited spatial accuracy and low sampling rates for reading analysis by heuristic fixation-to-word mapping

Oleg Špakov, Howell Istance, Aulikki Hyrskykari, Harri Siirtola and Kari-Jouko Räihä

Received: August 21, 2018/ Accepted: date

Abstract The recent growth in low-cost eye tracking systems makes it feasible to incorporate real-time measurement and analysis of eye position data into activities such as learning to read. It also enables field studies of reading behavior in the classroom and other learning environments. We present a study of the data quality provided by two remote eye trackers, one being a low-sampling-rate, low-cost system. Then we present two algorithms for mapping fixations derived from the data to the words being read. One is for immediate (or real time) mapping of fixations to words and the other for deferred (or post-hoc) mapping. Following this, an evaluation study is reported. Both studies were carried out in the classroom of a Finnish elementary school with students who were second graders. This study shows very high success rates in automatically mapping fixations to the lines of text being read when the mapping is deferred. The success rates for immediate mapping are comparable with those obtained in earlier studies, although here the data is collected some 10 minutes after initial calibration of low-sample (30 Hz) remote eye trackers, rather than a laboratory setting using high sampling rate trackers. The results provide a solid basis for developing systems for use in classrooms and other learning environments that can provide immediate automatic support with reading, and share data between a group of learners and the teacher of that group. This makes possible new approaches to the learning of reading and comprehension skills.

Keywords low-cost eye tracker · reading aid · fixation mapping algorithm · data quality · elementary school

O. Špakov
Tampere Unit for Computer-Human Interaction
University of Tampere
Finland
Tel.: +123-45-678910
E-mail: firstname.lastname@staff.uta.fi

1 Introduction

The advent of low-cost eye tracking systems brings new opportunities for using gaze measurement to support learning to read in schools, learning new languages as well as helping to comprehend unfamiliar or difficult passages of text. In addition to supporting the individual learner, data related to reading progress and reading difficulties can be shared between the learner and the teacher in close to real time. This enables the teacher to have an overview of the learners who are currently experiencing difficulties as well as those parts of the text being read that are causing problems to several learners.

For these benefits to be realized, the tracking system used needs to deliver usable gaze position data that is sufficiently accurate to enable identification of the words being read in close to real time or straight after reading, and to provide data about those words that are fixated upon. Furthermore, this performance has to be delivered during the whole of a normal school lesson, and accommodate the normal range of movement by the learner without any recalibration of the eye tracking device. The work reported here addresses means of overcoming limitations in the data quality provided by remote eye trackers that currently are cheap enough to be considered where multiple units would be bought by a school. These limitations arise primarily from the measurement accuracy and the sampling rate of the eye tracker. Needless to say, requiring learners to use some kind of head restraint would not be acceptable. There will inevitably need to be some automatic correction for calibration shift in the data obtained from the eye tracking system in order to map fixations to the words being read.

The paper presents a comparative study of data quality obtained from two eye trackers in a school classroom environment (Section 3). This is followed by a description of the mapping algorithms used. One of these attempts to map fixations to words being read in close to real time, while the other is used when all fixation data for a page of text has been collected. (Section 4). Section 5 then describes a study carried out in a classroom environment to evaluate the performance of both algorithms using a low-cost eye tracker with limited spatial accuracy and low sampling rate. Section 6 contains a discussion of the findings and Section 7 presents the conclusions of the studies.

2 Background

2.1 Previous Work

Typically reading behavior studies in the past have used high sampling frequency eye trackers that have required the participant to maintain a constrained head position, or to wear an eye tracking device on the head. Of central importance to the current work is how to monitor reading behavior using noisy data from low-cost eye trackers obtained from people whose position is not constrained, in a variety of indoor environmental conditions.

One way to approach the question of how to deal with noisy data is to try to apply a correction to the data as it is received from the eye tracker to take account of drift since the initial calibration was carried out.

Dynamic drift correction (Stampe and Reingold, 1995) involves correcting the data obtained from an eye tracker based on knowing where a person is looking at a specific time. The difference between the actual location of a target and the measured location of the gaze position can be used to correct data collected subsequently. Hornof and Halverson (2002) used ‘implicitly required fixation locations’ (or RFL’s) where the user’s visual attention could be expected to be at a particular time, in order to apply a dynamic drift correction to counteract systematic error. The locations were those of on-screen buttons that participants were required to click during an experiment.

Another approach in the context of reading is to treat all of the words displayed on a screen as ‘potential fixation locations’. The reader will look at all or most of these during the course of the reading task. If assumptions are made about how the text is read, then the spatial and temporal pattern of fixations can be mapped to the pattern of words on the screen. This is the approach adopted in this study and there are three broad areas of previous work that inform it.

The first is where an attempt is made to analyse reading behavior after it has been completed, using gaze data, for example, collected when reading web pages e.g. (Beymer and Russell, 2005). The second is where the analysis of words being read takes place in real time. It allows some kind of automatic action to be taken in response to detected events, such as assumed reading difficulties e.g. (Hyrskykari, 2006b). The third area of work is the study of the variability in the data obtained from medium- to low-cost eye trackers in field conditions. This is in anticipation of widespread use of gaze position as an additional input channel in the context of personal computing e.g. (Feit et al, 2017).

2.2 Post-Hoc or Deferred Mapping

If certain assumptions about reading behavior are made, then lines of text presented on a screen and the position of words within these lines represent potential fixation locations. The assumptions are: lines are read from left to right; almost every content word is fixated at least once; about 10-15% of saccades are regressions, usually within the same line; at the end of a line the reading point is transferred to the start of the next line; during reading (as opposed to skimming) transferring the gaze point to lines beyond the next line occurs rarely. (Hyrskykari, 2006b)

Beymer and Russell (2005) reported a post-hoc mapping system called WebGazeAnalyser intended to provide statistics on how people read web pages. They first grouped fixations into horizontal lines and then mapped these to lines of text extracted from the Document Object Model (DOM), which describes how the text being read is presented in the web browser. The mapping

took an estimate of the current gaze tracker drift into account. The grouping of fixations into lines used regression line fitting to the centers of consecutive fixations. Fixations were added to the current line while the fixation-to-line error remained within a certain tolerance. A similar approach is described by Cohen (2013) for the automatic correction of fixation locations collected in reading experiments. This involves first the automatic correction of the vertical coordinates of fixations, and the removal of outliers and ambiguous fixations in data collected from reading experiments. To facilitate this, the assumption is made that the standard deviation of vertical locations from the regression line are constant along the regression line, and that calibration error is relatively consistent across the display. Both of these assumptions are dubious (Hyrskykari, 2006a).

Abduln and Komogortsev (2015) present an algorithm for mapping fixations to passes that match the lines of text being read. The passes are formed by detecting return saccades to the beginning of the next line or to the beginning of the page. Their algorithm makes use of a number of metrics provided by a high-frequency eye tracker. It is applied to person verification in a user authentication context.

Sanches et al (2015) compared three algorithms for post-hoc matching of lines of fixations to lines of text that had been read. The first matched the count of fixations to the counts of words in the respective lines on the screen. The second used the horizontal distance between the first and last fixation in a line and tried to match this to the distances between the centers of the first and last words on each line. The third approach used the differences between the horizontal coordinates of the fixations in a line, and the horizontal coordinates of the centers of the words in each line of text. It uses a function that minimises this quantity for each line. The authors provide an evaluation that compares the allocations of lines of fixations to lines of text by human judges and by the three algorithms. The best performance was obtained by the third approach, where 60% of fixations were judged to be correctly mapped to lines of text.

An alternative approach is to ignore the reading process altogether and see the issue simply as mapping gaze points to the bounding boxes of the words displayed on the page. Martinez-Gomez et al (2012) use image registration to do this for both reading and skimming. This involves finding a spatial transformation of the source image (a collection of gaze sample points, not fixations) to fit the target image (a collection of word bounding boxes). Assuming that the drift is largely vertical then the transformation becomes a translation and a scaling in y only. The best of three different approaches to deriving the transformation parameters produced a performance of approximately 70% of that arising from a manual adjustment of the gaze data to fit the word bounding boxes in what the authors call ‘careful reading’.

2.3 Real-Time or Immediate Mapping

Sibert et al (2000) describe a gaze-assisted reading assistant that attempts to detect reading difficulties and to provide assistance to the reader by automatically reading the word aloud. This requires real-time analysis and mapping of fixations to words being read on screen. The system was aimed at students who had recognized reading difficulties. The assumptions described earlier about reading behavior are used to track the current line and word within the line being read, and horizontal fly-backs in gaze position indicate either a regression within the same line has taken place or the transfer of reading to the next line of text. The system allowed manipulation of font size, inter-word separation and inter-line separation in order to affect the likelihood of correct determination of these two possibilities.

The iDict system (Hyrsykari, 2006a) used gaze behavior while reading text in a second language to indicate difficulties a reader had in understanding specific words. Several candidate indicators of reading difficulties were investigated, and the total time a word was fixated upon, mediated by the frequency of occurrence of the word in the language corpus, was used. In response to these assumed difficulties, a translation of the word was presented automatically above the word where it appeared on screen (and a dictionary definition was also presented in a separate panel on-screen). Two automatic drift correction algorithms were employed to decide which line of text a new fixation should be associated with. Both used the idea of a box, or a mask, forward of the most recently mapped fixation to decide whether or not to map the new fixation to the same line. The first, *Sticky lines*, made the vertical size of the mask bigger as reading progressed across the line of text. The longer the sequence of progressive fixations, the more vertical drift was tolerated in either direction. The second, *Magnetic lines* was intended to compensate for a vertical shift in the location of fixations at the beginning of a new line. At the start of reading a new line of text, the average fixation-to-line distance of the previous line is calculated and carried forward to the new line. The starting location of the box or mask is centered at this vertical location rather than on the actual height of the new line. A manual drift correction performed by the reader using arrows keys was also implemented to move the current line being read up or down if the automatic correction had failed. The performance of the automatic correction algorithms was found to be partly dependent on the spacing between the lines of text. In the best case (using 1.5 line spacing and 11-point Verdana font), an average of 86% of the fixations were judged to have been correctly mapped to the lines being read.

Palmer and Sharif (2016) report an algorithmic approach to adjusting the location of fixations when reading source code. This type of reading is different from reading text, as source code is generally not read in a linear fashion, top to bottom. Fixations are formed into temporal clusters and then the location of the clusters are moved to fit areas of interest (AOIs) formed by lines of source code, and lexical tokens within the lines. The output of the automatic adjustment was compared with that produced manually by two human judges

working simultaneously. The automatic mapping produced an accuracy of 89% of the manual mapping for line AOIs (rather than token AOIs).

2.4 Reading or Skimming

The success of the automatic association of the individual fixations to the lines of text being presented for reading relies on assumptions made about gaze behavior during reading. If the person skim-reads the text, perhaps to get the gist of the text, then these assumptions will not apply, or only partially apply. Biedert et al (2012) were able to produce a classifier capable of differentiating between reading behavior and skimming behavior. An essential element in this distinction was the distance covered by saccades expressed in the number of characters. An average saccade distance of 10 characters or less suggests reading behavior, based on the word identification span (Rayner, 1998) while skimming covers all other gaze behaviors. The features of the saccade data that provided the best classifier performance were described as saccade forward speed and angularity, rather than a simple distance expressed in characters.

2.5 Data Quality and Unrestrained Use

As eye trackers become less expensive, the prospect of using gaze data in a far wider variety of contexts related to personal computing increases. This use will be generally unsupervised and for this to be successful, a wide variety of different users will need to be accommodated. The effective accuracy and precision of the data the eye tracking system can provide may be quite different from that claimed by the system manufacturers, as these quantities usually have been obtained from testing under laboratory conditions.

Both Reingold (2014) and Holmqvist et al (2012) have pointed to the lack of a standardized methodology for establishing and reporting the data quality obtained from eye trackers used in a particular experimental situation. This could lead to an over-reliance on the manufacturer's figures quoted for accuracy and precision. Reingold addresses the measurement of data quality with two methods based on an artificial eye so that the same stimulus is presented to different eye trackers for the purposes of performance comparison.

Feit et al (2017) investigated the variability in data quality in a study of 80 people using two remote eye trackers in two conditions representing differences in ambient lighting conditions. One condition was intended to represent natural light and the other artificial light. Participants were asked to fixate at 30 locations distributed horizontally and vertically on the screen surface. Accuracy was significantly worse in the vertical direction compared with the horizontal. There was considerable variation in the precision across the screen surface. There was a significant difference in the accuracy between the two eye trackers, but the more accurate tracker also suffered from a much higher data loss than the other. In the case of the more accurate tracker, gaze position

could not be estimated in 13.1% of the targets fixated on, while the corresponding figure for the other tracker was only 2.8%. There was no significant difference between the two lighting conditions.

Niehorster et al (2018) investigated performance differences between five remote eye trackers especially in relation to recovery from loss of the image of one or both eyes, say when a person turns their head and looks away from the screen. During tests, the head was rotated away from the normal front facing view, and one or both eyes were hidden temporarily from the tracker's cameras. The effect of these events and recovery from these on the data obtained from the eye trackers was studied. There were considerable differences between the systems tested in the amount of data loss incurred and offsets between the measured position and actual position of the target being viewed. The study also provides a test methodology for examining the performance of low cost remote trackers being considered for use as input devices for computing systems.

In the next section, we report a study of the quality of data obtained in a field study from two remote eye trackers, one of which is the system intended for use with the reading aid.

3 Study of Eye Tracking Data Variability

In order to build a gaze-enabled reading aid for use during normal lessons in a school classroom, it is necessary to study the quality of data that can be obtained from low-cost eye trackers under realistic conditions. This will enable the font size, text line length, and inter-line separation of the text to match the performance of the eye tracker. If the font size or line separation required are too great to produce a usable and attractive screenful of text to be used during a lesson, then the feasibility of a gaze-enabled reading aid is called into question, at least with current eye tracking technology.

The system needs to tolerate normal movement and sitting posture of the student during lessons, and to be calibrated only once at the start of the lesson. Data of acceptable quality then needs to be available over the course of a 45-minute lesson.

A study of 23 second grade students, aged between 8 and 9, was conducted in the classroom in a Finnish elementary school. The performance of two remote eye trackers was compared as described in the next section. The School's standard procedure for obtaining parental consent for the participation of the students in classroom-based research activities was followed. Information about the project was sent home with a student. This explained the purpose of the study, that the student need not participate, could withdraw from the study and that all collected data was anonymous. Additionally, during the data collection trials, the teacher asked for volunteers amongst those who had not yet participated. All those present chose to do so. The names of the students were not provided to the researchers.

3.1 Equipment

Two Dell E7520 laptops with a screen resolution 1366×768 and a 12.5 inch screen were used, one with a Visual Interaction myGaze n, produced by SMI, which was the low-cost eye tracker intended for use with the reading aid. This was a remote binocular tracker with a sampling rate of 30 Hz. According to the manufacturer¹ this had an accuracy of 0.4° and a spatial resolution of 0.05° . The other was used with a Tobii X2-60 eye tracker, which was included in the study for comparative purposes. This binocular tracker had a sampling rate of 60 Hz. The claimed performance² was an accuracy of 0.4° and a spatial resolution of 0.34° . The Tobii EyeX low-cost remote tracker was considered for inclusion in the study but the manufacturer's licence conditions prohibited this.

3.2 Data Collection Procedure and Task

Both systems were set up on adjacent desks in the classroom close to the wall away from the windows, facing into the room. The other desks in the classroom were arranged as usual in clusters of four with the students facing each other.

Students carried out the tests two at a time. This was at the suggestion of the teacher in order to reduce the feeling that it was the students as individuals that were being tested. After the tasks were completed, the students swapped over and carried out a similar task with the other eye tracker and laptop. Half of those tested used the myGaze system first, and half the Tobii system first.

The tests using a target identification task were conducted simultaneously. First, there was a 9-point calibration sequence supervised by a member of the research staff. Then the identification task was conducted for the first time. This was followed by a 10-minute period where both students played a number of on-line Finnish language and reading games chosen by the teacher. Then a similar identification task was carried out but with no re-calibration. After this, the students swapped seats and the procedure was repeated. The purpose of the 10-minute game playing session was to simulate a period of normal lesson activity.

During the target identification task, targets were displayed one at a time in random order at one of the 16 positions shown in Figure 1. Each target was a combination of different spaceships and flags of countries bordering the Baltic Sea. Each target was presented for 1.8 seconds. The task was to count the number of spaceships belonging to a particular country, e.g. 'count the number of Swedish spaceships'. The intention was to keep the participant's gaze fixated on a location on the screen just long enough to enable fixation data to be collected. The target size was 48 pixels horizontally \times 24 pixels

¹http://www.mygaze.com/fileadmin/download/mygaze_n/Technical_Specification_myGaze_n.pdf

²<https://www.tobiipro.com/siteassets/tobii-pro/technical-specifications/tobii-pro-x2-60-technical-specification.pdf?v=1.0>



Fig. 1: Targets and locations used in target identification task

vertically. This was chosen to represent the dimensions of a word of 4 characters displayed in 18pt Arial font.

Target locations were partitioned into central and peripheral locations, indicated by the dotted line in the figure, not visible during the tests. Pilot tests indicated that the test was too long when displaying targets in all 28 row-column intersections and students were losing motivation towards the end of the test. Consequently, 12 of the peripheral locations were removed.

3.3 Results of the Classroom Tests

Although 23 students took part in the trials, a change was made to the static vertical calibration parameter of the myGaze system after 14 participants had been tested. A systematic vertical offset was observed in the recorded data, which led to a change being made in the specified distance from the tracker to the bottom of the screen. The data presented here are for the nine participants tested after this change was made. There are two main features of interest in the data: the extent of variation in fixation locations after the period of simulated class activity for each tracker; and the number of targets for which no data was collected, again for each tracker.

3.3.1 Offsets in measured fixation location in relation to the target position

Figure 2 shows the vertical and horizontal offsets of the measured fixations from the centers of the targets, for both data collected immediately after calibration (START), and after 10 minutes of classroom activity (END). The data points in the figure are the absolute values of the offset. The raw data

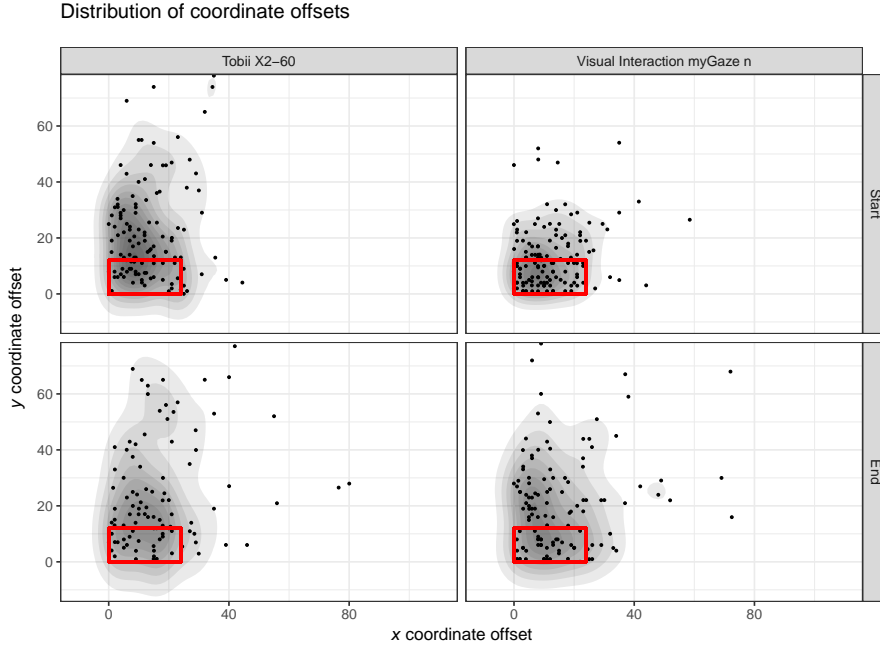


Fig. 2: Absolute vertical and horizontal offsets of coordinates for two trackers at the beginning of the experiment (START) and after 10 minutes of classroom activity (END). The distributions of coordinates are shown as a 2D histogram overlay (2D density plot), and the target is shown as a red box. The coordinate offsets and the target box are in absolute values.

is overlaid with a 2D histogram to show the coordinate distributions, with a red box representing the target. Within each target (24 x 48 pixels), fixations were measured as offsets from the centre, and were either positive and negative. As these are shown as absolute values in figure 2, the red box size is 12 X 24 pixels. The outliers above value 110 from y -coordinates are not shown to improve readability, but are included in the computations. The data contains both central and peripheral targets.

Figure 3 shows the distribution of coordinate offsets as boxplots. The data is divided to central and peripheral targets (Figure 1), and again the outliers above value 110 from y -coordinates are not shown to improve readability, but are included in the computations.

In the tests immediately following calibration, the extents of the vertical offsets for the X2-tracker are greater than those for the myGaze-tracker. This difference results from the central targets rather than the peripheral targets. The median and 75th %-tile for the y -coordinates of the offsets are twice as large for the X2-tracker as for the myGaze-tracker. The medians and interquartile ranges for the x -coordinates of the central targets are very similar between

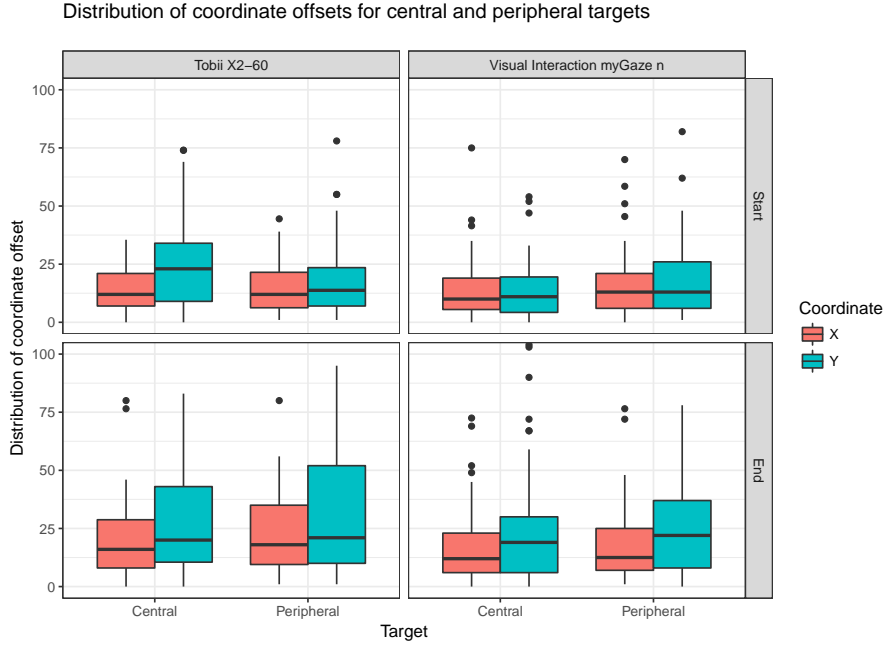


Fig. 3: Absolute vertical and horizontal offsets of coordinates as boxplots. The data are shown separately for the two trackers at the beginning of the experiment (START) and after 10 minutes of classroom activity (END), and for the central and peripheral targets (Figure 1).

the trackers. The ranges and medians of the offsets in both x and y for the peripheral targets for which data was collected were very similar between the trackers.

After the 10 minutes of classroom activity, the differences between the trackers have reduced. There are still differences for the 75th %-tile for the y offsets for the central targets although the medians of these between the trackers are now very similar. The median values of the offsets in the x direction are well within the bounds of the target (24 pixels) for both trackers even after 10 minutes of activity following calibration. For both trackers, the medians of the y -values fall outside the target bounds (12 pixels). For the myGaze tracker, the medians for the central and peripheral targets are 19 and 22 pixels respectively. For the X2-tracker, the corresponding values are 19.7 and 21.5 pixels.

3.3.2 Comparing missing data rates between the two trackers

The big difference between the trackers however is in terms of data loss (Table 1). There is no data loss with the myGaze-tracker for either central or

Table 1: Data loss arising when 9 participants viewed 10 central targets and 6 peripheral targets using the 2 trackers at the START and after 10 minutes (END) , n = number of valid data sets.

		<i>myGaze</i> n			<i>Tobii X2-60</i>	
		Total	n	% data loss	n	% data loss
START	Central	90	90	0.0	80	11.1
	Peripheral	54	54	0.0	47	13.0
END	Central	90	84	6.7	70	22.2
	Peripheral	54	49	9.3	42	22.2

peripheral targets in the test immediately following calibration, while the X2-tracker did not record data for 11% of the central targets and 13% of the peripheral targets. There is data loss for both trackers after 10 minutes activity, although the rates for the X2-tracker are considerably higher. For central targets the loss rates for myGaze-tracker and X2-tracker are 6.7% and 22.2% respectively, while for peripheral targets the corresponding values are 9.3% and 22.2%.

3.4 Observations Made During Trials in the Classroom

Most of the 23 children sat very still at their usual desks during the parts of the lesson where they had been told to read or to write and appeared to focus strongly on the tasks they had been given. Two or three children changed sitting postures frequently at their desks, and were similarly mobile during the 10-minute activity and the second eye tracking tests.

Some of the students sat relatively high above the desk surface, including one who sat on a large rubber ball³. This would require the laptop screen with the attached eye tracker to be positioned with a shallow angle to ensure that the student's eyes were within the head-box of the tracker. There would be a greater risk of reflections of the ceiling lights behind the student appearing in the screen than if the sitting height of the student was less. The tests were carried out with chairs from the classroom chosen because of their height.

The desk area was relatively small, which suggested that the computer used to support the reading aid would have to be moved out of the way during periods of handwriting, and then repositioned for use during reading. This could have significant implications for the maintenance of the initial calibration.

The calibration itself would need to be carried out by the students without supervision as the teacher would not have time to do this. The use of a game, designed around a specific calibration game mechanic, has been shown to be an

³<https://www.livestrong.com/article/287638-pros-cons-of-sitting-on-a-stability-ball/>

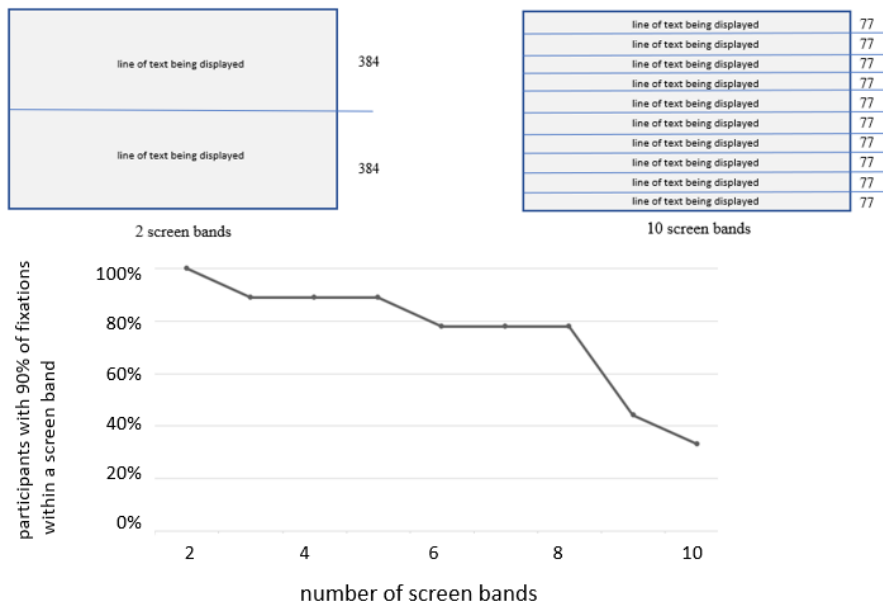


Fig. 4: Percentage of the 9 participants with 90% of their fixations on central targets falling within screen bands of different heights

effective approach to this issue. The quality of calibration data obtained from unsupervised second grade children using such a game compared very favorably with that obtained from supervised adults using the standard moving target calibration procedure.(Špakov et al, 2018)

3.5 Mapping Fixation Dispersion to Text Line Separation

The vertical offset data shown in Figure 3 can be used to derive a separation between lines of text on the screen based on the extent of the region above and below the target where a large proportion of the fixations occur. The screen which was 768 pixels high (y -extent) can be divided into equal horizontal bands as shown at the top of Figure 4, from 2 bands each 384 pixels high, to 10 bands each 77 pixels high. For each size of screen band, the percentage of participants with 90% of their fixations on central targets falling within the band is shown. All nine participants (or 100%) had 90% of their fixations falling within a vertical range of 384 pixels from the target center (corresponding to two screen bands). Only three (or 33%) had 90% of their fixations falling in a range of 77 pixels (corresponding to 10 screen bands).

If the target was to be replaced with a line of text 24 pixels high (the same height as the spaceship), then in the first case there would be 2 lines of text

on the screen, and in the second case there would be 10 lines of text on the screen.

Setting the inter-line separation to ensure 90% of fixations made by 100% of participants can be correctly associated with the line being read means that it will only be possible to display two lines of text on the screen at a time. This is not likely to be acceptable. However, if associating 90% of the fixations for approximately 90% of participants is acceptable, then the number of the bands increases to five. In other words, five lines of text could be displayed on the screen and 90% of fixations could be correctly associated with the line being read for 90% of users only on the basis of spatial proximity. This is using data provided by a 30 Hz tracker with unrestrained users. If, in addition, a means of intelligently mapping of fixations to the lines of text is applied, then inter-line separations corresponding to more than 5 screen bands should be possible with same quality of data.

In summary, the study of data quality established that the data obtained from the myGaze low-cost eye tracker was capable of delivering data from unrestrained users in a classroom environment for a reasonable period after initial calibration. Data loss for peripheral targets was less than 10% and for central targets it was 6.7%. The 75th %-tile of the offset in the x -direction lay close to the bound of the target area (24 pixels from the center of the target). These values were 23 pixels for central targets and 25 pixels for peripheral targets. The performance of the tracker in the y -direction was less good, and the medians fell outside the target bound of 12 pixels. However, the approach to fixation-to-line mapping emphasizes computing the likely line of text being read, in other words the y -coordinate of the target being fixated upon. Thus it was expected that the mapping algorithms used would compensate for the lack of accuracy in the y -direction.

The next section describes the algorithms used to map fixations on lines of text and words within the lines.

4 Fixation Mapping Algorithms

It is appropriate to distinguish between immediate mapping and deferred mapping of fixations to the words being read in the text. Immediate mapping means making a decision when the fixation is detected about the word it is associated with based on the data already collected and knowledge of the text being read. This is necessary if some automatic action is to be taken in real time or close to real time, such as displaying some additional information to facilitate the reading of that word. Deferred mapping means waiting until fixation data is available for all of the words that have been read, and then fitting the fixations to individual words. The algorithms used for each are different.

4.1 Definition of a Fixation

When eye trackers are used that sample at say 1000 Hz, or at 1 ms intervals, a definition of what constitutes a saccade in terms of the distance between consecutive samples is based on velocity and is fairly unambiguous. A fixation is anything that is not a saccade. Gaze position was in this study sampled at 30 Hz, which is a relatively low sampling rate. As samples are collected at 33 ms intervals, accurate determination of individual saccades is not possible, and it is necessary to define what constitutes a fixation.

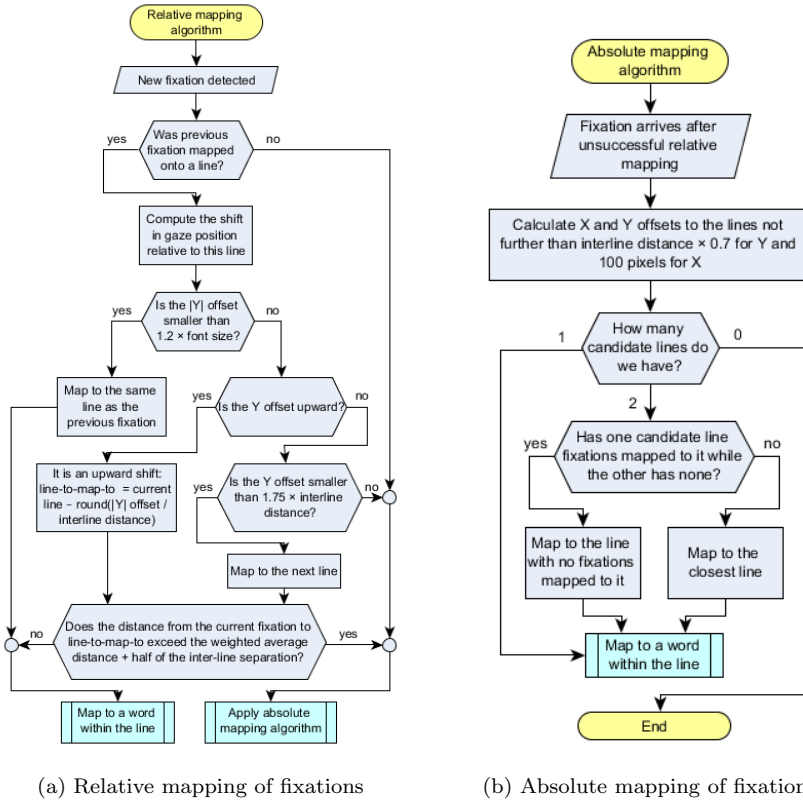
In this work we consider that a gaze sample just arrived from the eye tracker continues an ongoing fixation only if it is close enough to the center of this fixation (not further than 70 pixels). In this case the gaze sample is aggregated with the current fixation. The new center of the fixation is calculated as $old\ center \times 0.4 + gaze\ point \times 0.6$. The coefficients were arrived at by experimentation; the coefficients that worked best with typical data were used. If the gaze point is further than 70 pixels from the center of the current fixation, it becomes a candidate to start a new fixation, but the current fixation is still recognized as possibly continuing. The current fixation is considered finished only if two consecutive gaze samples are located far enough from its center. The fixation detection algorithm produces a new fixation event only when there are three consecutive gaze samples close enough to each other, taking 100 ms to collect. This kind of fixation detector with noise filter is used often in applications with low-sample gaze data e.g. (Sharmin et al, 2013). For a full discussion of fixation detection, see Holmqvist et al (2011, chap. 5.5).

4.2 Immediate Mapping

The information that is available to map fixations to words is the position of lines of text on the ‘page’ or screen being read and positions of the individual words within a given line. Information is also available about the fixations that have been previously mapped on the same page. No mapping information is carried over between pages. There are two algorithms for immediate mapping: *relative* mapping, where the previous fixation has been mapped to a word and this information is used to map the current fixation; and *absolute* mapping, where no reference is made to the previous fixation. This builds on previous work by Hyrskykari (2006a). An overview of the main steps in the process of relative mapping is shown in Figure 5a and in the process of absolute mapping in Figure 5b. A description of both of these algorithms as pseudo code is given in Appendix B and Appendix C respectively.

4.2.1 Relative mapping

In Figure 6, the fixation that is currently being mapped is shown between the first and second line of text. If the previous fixation has been mapped to a line of text, then the algorithm tries to compute the shift in gaze position relative



(a) Relative mapping of fixations

(b) Absolute mapping of fixations

Fig. 5: Immediate mapping of fixations to lines of text being read

to this line. Four fixations where the mapping is considered to be *reliable* have been mapped to the first line. For a fixation’s mapping to be considered reliable, it may not be the first fixation mapped to that line, and it should be a progressive fixation (its x -coordinate is to the right of the previously mapped fixation). These have been labeled ‘R’ (and ‘NR’ for *non-reliable*) in Figure 6. Note that in Figures 6, 7, 8, and 11 the gaze path is hypothetical and does not represent actual collected reading data; the figures are intended to highlight the parameters used in the algorithm.

The difference in the vertical coordinate of the new fixation from the previously mapped fixation is compared against two threshold values. The first is the maximum change that would still indicate that the same line is being read. Currently, this is set to $font\ size \times 1.2$ (smaller box in Figure 6).

The second threshold is what the maximum shift downwards of the gaze position would be if the next line was being read. Currently this is set to $inter-line\ separation \times 1.75$ (larger box in Figure 6). If the vertical shift is less than this (and greater than the first threshold), it is assumed reading has moved to the next line, as is the case in the figure. Thus the fixation would be

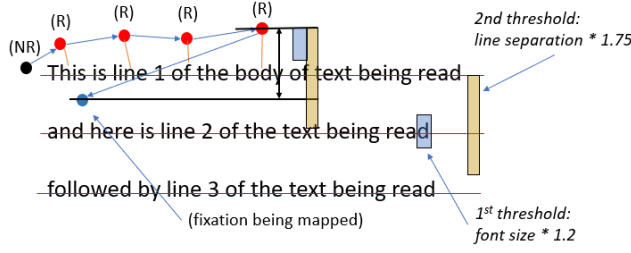


Fig. 6: Mapping the line being read relative to the previous fixation

mapped to the second line, but the mapping would not be considered reliable as this is the first fixation mapped to the line.

If the shift is greater than both thresholds, then the absolute mapping algorithm is applied (see Section 4.2.3).

A regressive shift upwards is indicated by a negative change in the vertical coordinate. The value of this change is divided by the inter-line separation. The rounded result of the division is taken as the change in lines upwards from the current line.

4.2.2 Making an additional check on intended mapping

In all cases, an additional check on the line that the fixation will be mapped to is then made. The average of the distances of previously mapped fixations to their respective lines is calculated, each average being weighted by how far its line is from the fixation currently being mapped. If the distance from the current fixation to the line it will be mapped to exceeds the weighted average + half of the inter-line separation, then the result of the relative mapping is considered unreliable and is discarded. Absolute mapping is used instead to determine the line associated with the current fixation.

The average fixation-to-line distance (or offset) for previously mapped fixations is computed for each line using only reliable fixations.

$$\text{Weighted average} = \frac{\sum(w_i L_i)}{\sum(w_i)} \quad \text{where } w_i = \sqrt{\frac{100}{D_i}}$$

L_i is the average fixation-to-line offset of line i , w_i is the weight given to L_i , and D_i is vertical distance of line i to the fixation being mapped. The minimum value of D_i is taken as 1.

In Figure 7, the distances for the two lines with fixations mapped to them will be

$$L_1 = \frac{(a + b + c)}{3} \quad \text{and} \quad L_2 = \frac{(d + e + f)}{3}$$

The weight factor w_i is an arbitrary amount, intended to give greater importance to the average fixation-to-line offsets that are closer to the fixation

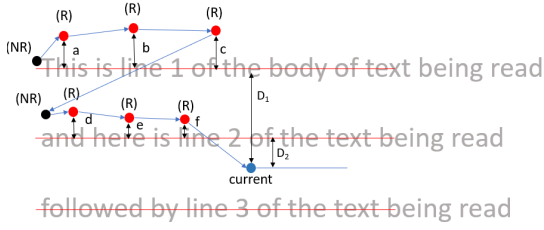


Fig. 7: Calculation of the weighted average fixation-to-line offset

currently being mapped. Once the distance of a line with previously mapped fixations to the current fixation exceeds 100 pixels, then the weight given to its average fixation-to-line distance (L_i) decreases rapidly. The maximum weight given to an average fixation-to-line offset is 10, when the distance D_i from that line to the fixation being mapped is 1, the minimum allowed.

Applying the rule for relative mapping to the case shown in Figure 7, the third line of text would initially be selected for the fixation being considered. This is on the basis of the vertical distance from this to the previously mapped reliable fixation falling between the 2 thresholds shown in Figure 6. However, if the distance from the current fixation to the third line exceeds *weighted average* + ($0.5 \times \text{inter-line separation}$), the mapping to the third line would be considered unreliable, and absolute mapping would be applied instead.

4.2.3 Absolute mapping

This refers to immediate mapping of a fixation to a line of text being read, without reference to, or knowledge of, the line that the previous fixation was mapped to. The algorithm has knowledge of the vertical coordinate of each line center, the horizontal extents to the left and right of the words on a particular line, and of reliable fixations previously mapped to each line. (see Figure 5b and Appendix C)

First, the vertical distance from the fixation being mapped to each line is calculated. If the fixation is left of the boundary of the leftmost word, or right of the boundary of the rightmost word, the horizontal distance from the boundary is also calculated. Next, lines where the vertical distance exceeds the *interline distance* $\times 0.7$ or where the horizontal distance, left or right, exceeds 100 pixels, are dropped from the list of candidate lines for mapping the fixation to. This will leave either 0, 1 (upper or lower line) or 2 lines (upper and lower line) as candidates. If there are no candidate lines, the fixation is not mapped at all. If there is one candidate, then that is the line the fixation is mapped to. If there are 2 candidates, the decision of which to choose is based on the following procedure.

The normalized distance score (NDS) from the fixation to each of the lines is the sum of the normalized vertical and horizontal distances, and has a value

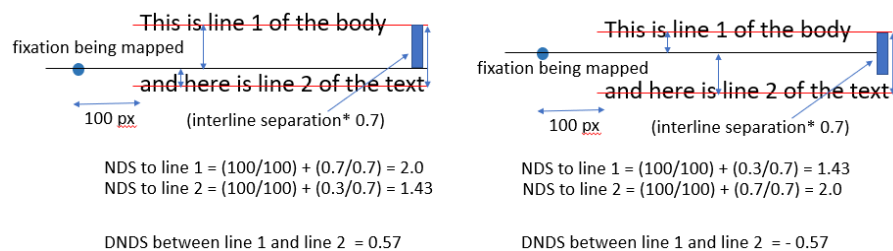


Fig. 8: Calculating the Difference in Normalized Distance Scores (DNDS) for line selection in absolute mapping

from 0 to 2. The normalized vertical distance is distance from fixation to line divided by *interline distance* $\times 0.7$ (see Figure 8). The normalized horizontal distance is the extent of the horizontal location in pixels of the fixation outside the left or the rightmost word boundary of a line divided by 100. This takes account of differences between the horizontal extents of candidate lines for mapping, as may often be the case on the right hand edge of the text. This may also be the case of the left hand side if there is a left indent of one of the lines. The difference in normalized distance scores (DNDS) to both candidate lines is calculated.

When both lines already have reliable fixations mapped to them, then where the DNDS is negative the fixation is mapped onto the upper line. It is mapped to the lower line where the DNDS value is positive, a DNDS value of 0 being the threshold. Otherwise the algorithm favors the line without reliable fixations already mapped to it. This is done by moving the threshold towards the line that has fixations mapped to it by an arbitrary amount (0.3) that was determined from initial testing. If there were reliable fixations mapped to 'line 1' but not 'line 2' of Figure 8, the threshold would be adjusted to -0.3. The fixation shown in the right hand side of Figure 8 with a DNDS score of -0.57 would still be mapped to 'line 1', but a fixation with a DNDS score of more than -0.3 would be mapped to 'line 2'.

4.2.4 Mapping fixations to words within a line

If the fixation has been mapped to a line of text either through relative mapping or absolute mapping, then an attempt to map it to a word within the line is made. By default the fixation is mapped to the closest word in the line by considering the horizontal coordinate of the fixation in relation to the left and right boundaries of each word. This mapping is not made if the horizontal distance of the fixation is greater than 100 pixels from the left or right boundaries.

However, if a word already has fixation(s) mapped to it, or if the word length exceeds a threshold of 5 characters, then fixations where the horizontal coordinate aligns with its last 3 letters will not be mapped to this word, but

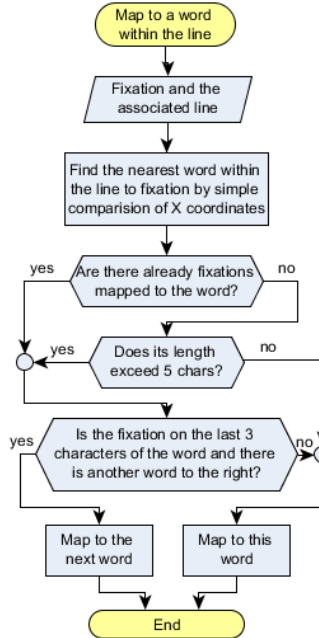


Fig. 9: Mapping fixations to words within a line of text

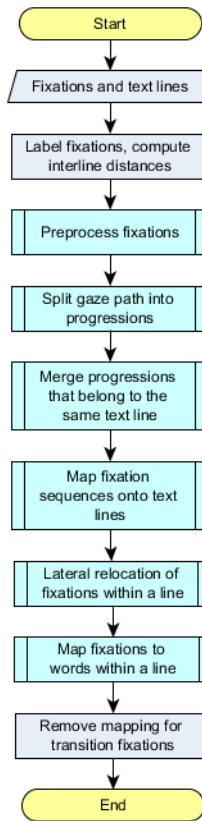


Fig. 10: Overview of deferred mapping process

instead to the next word, if one exists. This correction was implemented based on multiple observations of horizontal shifts in the data, when fixations landing onto the end of one word clearly belonged to the next word. The rationale for this is based on the asymmetric reading span of a fixation that extends 14 to 15 characters to the right of the fixation but only three to four characters to the left of it (Rayner, 1998). In view of this, a fixation falling at the end of a word that already has fixations associated with it is more likely to be made when reading the next word in the line. (see Figure 9)

4.3 Post-Hoc or Deferred Mapping

The second approach to mapping fixations to words waits until the reader indicates in some way that they have finished reading a complete screen. The general approach is to form sequences (or *progressions*) of fixations that are

made when reading the individual lines of text, and then map each progression to a line of text.(see Figure 10) and Appendix D)

The algorithm involves the following steps.

Data preparation. Fixations are labeled, and words are given an index within the body of text and within each line. Also the median line height and median line-to-line distance are calculated. Although the line-to-line distances will be generally constant, the distance from header to text lines, and between paragraphs may be different.

Fixation pre-processing. An attempt is made to merge very short-duration fixations with adjacent fixations if these are close enough to them. Alternatively, these are removed from the sequence if there are no near fixations. Optionally fixations too far from the nearest word may also be removed at this stage. These may be fixations made before or after reading the text, or when glancing away from the text during reading.(see pseudocode in Appendix D.1 and flowchart in Figure 14(left), Appendix A)

Splitting fixations into progressions. Here the whole sequence of fixations is split into a collection of shorter sequences. Each new sequence contains fixations that belong to a continuous sequence of progressive reading. If a fixation lies within a specified region in relation to the preceding fixation, it is added to that sequence, otherwise a new sequence is started. Figure 11 shows the region defined by four parameters. P2 is the horizontal extent of the region forward of the previous fixation. P3 is half of the vertical extent of the rectangular region. P1 extends the region slightly leftwards. It has been observed that some fixations in a progressive sequence are made slightly to the left of the preceding fixation, although fixating on a previous word in the text would be considered a regression. P4 defines the angle that the region can expand upwards and downwards to the right of the fixation. This is similar to how Hyrskykari (2006a) implemented the Sticky Lines approach to the real-time mapping of fixations to lines of text in the iDict system. (see pseudocode in Appendix D.1 and flowchart in Figure 14 (right), Appendix A)

Merging progressions that belong to the same text line. Next, sequences of fixations that are considered to be close enough to each other to be made when reading the same line of text are merged.

This involves combining fixation sequences in pairs and estimating a linear regression model for the combined data set. Then the pair that produces the smallest fitting error is merged into one sequence, replacing the two original sequences, and all other merging results are discarded. However, no merging occurs if the fitting error term e and gradient a do not lie within acceptable threshold values. This process is repeated iteratively until there are no more pairs of sequences that can be merged. Figure 12 shows an example of merging two sequences, where the linear regression model parameters of the combined sequence and the least squares error of fitting are shown with the subscript c . If these two sequences are merged, then the possibility of merging the new combined sequence with the sequence on the center of the figure would be tested on a subsequent iteration.

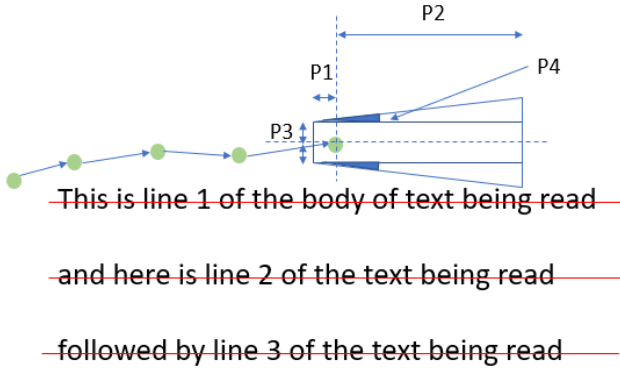


Fig. 11: Area for identifying fixations within a progression during deferred mapping

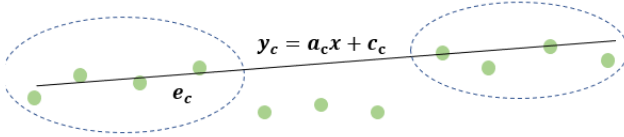


Fig. 12: Merging progressive fixation sequences during deferred mapping

The merging process is split into several phases. In the first phase, only sequences that contain at least 3 fixations are considered. When there are no more pairs of sequences that meet the criteria of length and satisfy the threshold values of error and gradient in the combined regression model, a second phase of merging is carried out. In this, all sequences regardless of the length are included, but short sequences (less than 3 fixations) can be combined only with long sequences. When no more mergers are possible under these conditions, the process is repeated again. In this third phase, no restrictions are placed on the length of sequences that can be merged. If after the third phase there are more fixation sequences than lines of text, then the merging process is repeated once again without applying thresholds to the fitting error term e and gradient a in the combined sequence. (see pseudocode in Appendix D.2 and flowchart in Figure 15 (left), Appendix A)

Mapping fixation sets onto text lines. At this stage, the number of sequences should not exceed the number of lines of text, and sequences following a regressive fixation will have been merged with the original progressive sequences. The order and timestamps of fixations within each sequence will have been preserved even after merging. The merged sequences are then sorted in order of their average y -coordinates. The topmost sequence is mapped to the top line of text, unless the sequence is notably longer than the line of text. Short titles often do not accommodate enough fixations to constitute a ‘reading’ sequence.

In this case, the topmost sequence is mapped to the first line long enough to accommodate these, which is usually the line following the title. Then each sequence is mapped onto successive lines of text.

When this is done a check is made by comparing the distance between the average vertical coordinate of fixations in this and in the previous sequence, with the average distance between the line centers. If the difference between the averages of sequence vertical coordinates (D) is greater than 1.7 times the average line separation, a correction in the mapping is made. This involves moving the current sequence of fixations to a line of text beneath the current line of text being considered. The mapping of the sequence of fixations is advanced by $(D/\text{inter-line separation}) - 1$ lines. (see pseudocode in Appendix D.3 and flowchart in Figure 15 (right), Appendix A)

Relocation of fixations within a line. At this stage a sequence of fixations has been mapped to a line of text. The algorithm will then try to adjust the horizontal position of the fixation within the text line if necessary. If the sequence’s leftmost fixation is to the left of the end of the first word in the line, and/or if the rightmost fixation is to the right of the last word in the line, then a correction to the horizontal position is made. The leftmost fixation is moved horizontally to the third letter of the first word. The rightmost fixation is moved horizontally to the sixth letter from the end of the last word, but no closer to the beginning of the last word than the third letter. The other fixations are re-located by scaling their x -coordinate accordingly. The extent of this horizontal scaling is, however, limited to 10%. That means the distance between the first and last fixations can only be reduced to 90%, or extended to 110% of the original. (see pseudocode in Appendix D.4 and flowchart in Figure 16 (left), Appendix A)

Mapping fixations to words within a line. At this stage a sequence of fixations has been mapped to a line of text and the horizontal coordinates have been adjusted based on knowledge of the horizontal extents of the words in the line. Next the algorithm goes through the sequence of fixations and maps these to words using the horizontal coordinate of the fixation. This uses a direct (naïve) mapping, although the fixation’s position is tested against the ‘effective’ word width rather than the actual width. This is computed as a fraction (0.7) of the word width in pixels if this exceeds a specific threshold width. The same effective width is also used if the word has fixations already mapped to it. If neither of these cases is true, the effective width is equal to the actual width.

If the fixation’s horizontal coordinate lies between the left bound of a word and the left bound + effective word width, the fixation is mapped to it, otherwise the fixation is mapped to the next closest word. (see pseudocode in Appendix D.4 and flowchart in Figure 16 (right), Appendix A)

Removing transition fixations Some fixations, while being part of the reading process, should not be mapped to any word. These are fixations that are classed as ‘transition’ fixations and occur during corrective saccades made while moving the gaze position from the end of one line to the beginning of another. Hyrskykari (2006a, Section 8.5) discusses transition fixations in de-

tail. Essentially, these arise when the return sweep to the start of the next line is too short, and a corrective saccade is made leftward to the first word in the line. The algorithm identifies these fixations and disassociates (un-maps) these from any word.

The next section describes a reading study carried out in the classroom to evaluate the performance of the algorithms described in this section.

5 Evaluation of Algorithm Performance

The spaceship test described in Section 3 had demonstrated that the low cost tracker was capable of delivering usable gaze data at least ten minutes after initial calibration. In Section 4, algorithms to compensate for the inaccuracy arising from the unconstrained sitting position of the reader were described.

In order to test the effectiveness of the two mapping algorithms (immediate and deferred), a set of user trials was carried out in the same class with the same students that had participated in the spaceship test. The intention was to conduct the trials in similar conditions in the classroom to those in a reading class in which the gaze-enabled reading aid could be used.

5.1 User Trial Data Collection Procedure

Text about asteroids and celestial objects, chosen by the class teacher, was presented on three screens. The screens contained 4, 5 and 6 lines of text respectively. Each screen of text used one of five different line separations. The largest line spacing corresponded to 6 screen bands, and the smallest line spacing corresponded to 12 screen bands (see Section 3.5). Only the myGaze n eye tracker was used. All pupils in the class volunteered to take part and 24 data sets were collected. The trials were conducted with one student at a time. The same researcher conducted these trials as had conducted the spaceship trials and testing students two at a time was not thought to be necessary.

The eye tracker was calibrated under supervision and then the students were asked to play educational games for ten minutes. As before, these were language and reading games chosen by the teacher. There was no subsequent recalibration. After this, the student was asked to read the first screen and tell the researcher when they had finished. She saved the data and caused the next screen to be displayed, and the process was repeated.

5.2 Fitting Fixations to Lines of Text

Of the 24 participants, data from two participants were excluded from the analysis (a total of six screens), and in the case of another three participants data from two of the three screens read were excluded from the analysis, leaving 60 complete data sets out of a possible 72. The reasons for these exclusions are discussed in Section 6.2.

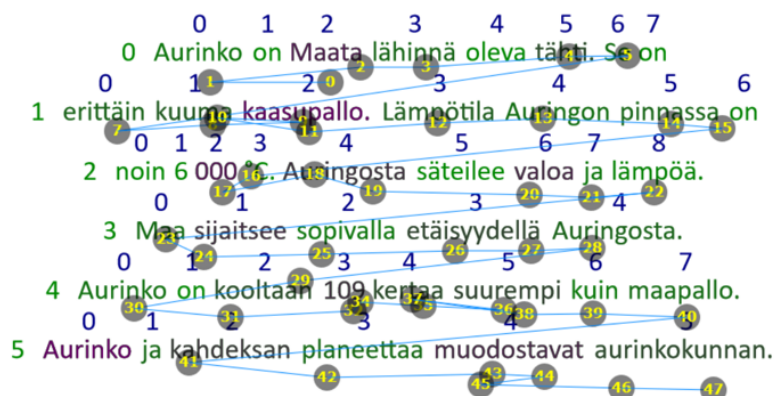


Fig. 13: Example of the locations of fixations in relation to text lines (screen 3, smallest line separation)

Both the immediate and the deferred mapping algorithms first attempt to allocate a recorded fixation to a particular line of text being read. Then if this is achieved, the fixation is allocated to a word within that line. Most of the computational effort in both algorithms is devoted to mapping a fixation to a line of text. The analysis of the data examines the level of agreement on the allocation of fixations to lines between two human judges and the immediate and deferred mapping algorithms respectively.

For each of the screens included in the analysis, the fixations were labelled automatically and presented overlaid on the text. On each screen, two human judges independently classified each of the fixations as either being made during reading, or outside reading (i.e. before reading started, or after the reading had been completed). Of the fixations made during reading, they then allocated these to the line of text that it appeared that the participant had been reading. Alternatively, a fixation was classified as a transition fixation. An example is shown in Figure 13. This shows the third screen read by one of the participants and contains 6 lines of text, labeled 0 to 5. In this case the smallest of the five line separations has been used. In this case there would have been space for 12 lines of text on the entire screen.

Here the participant made 47 fixations, and the two judges were in agreement about which line to allocate these to in all but one case. This was the fixation labeled 16, one judged this to be made reading line 2 while the other judged it to be a transition fixation.

The performance of the immediate mapping algorithm and the deferred mapping algorithm was assessed for each fixation. The mapping of a fixation to a line produced by each algorithm was judged a success if it agreed with at

least one of the two judges. The proportion of the number of successes to the total number of fixations made was taken as the score for each algorithm for that screen read by that particular participant. In the case of Figure 13 the score for the deferred mapping algorithm was 97.8% and for the immediate mapping algorithm 28.3%. The reason for the low success rate of the immediate mapping was that fixations 0 to 5 were mapped to line 1 and not to line 0. Then until fixation 29, fixations were allocated to the line below the one judged by the human judges to be the one being read.

Both judges thought that fixation 0 was a transition fixation and that fixation 1 was made reading line 0. The deferred mapping algorithm mapped fixation 0 to line 0, but then produced mappings that accorded with those made by at least one of the judges. The judges produced different decisions for fixation 16, but the deferred mapping algorithm agreed with one of them.

The median of the agreement scores for each of the mapping algorithms is shown in Table 2.

Table 2: Median of agreements between human judges and algorithms when allocating fixations to lines read

Agreement between two judges	Number of screens	IMMEDIATE mapping	DEFERRED mapping
100%	37	92.9%	96.2%
99.9 – 95%	17	92.3%	96.9%
95% or less	6	88.4%	94.4%
Total	60		

The 60 screens evaluated have been divided according to the extent of the agreement between the two judges. In 37 cases of the 60, the judges were in complete agreement on each of the fixations made on the screen. The median level of agreement between the deferred mapping and both judges (as they were in complete agreement) was 96.2%. For the immediate mapping the corresponding value is 92.9%.

The other two rows contain the screens where there was some disagreement between the judges. The deferred mapping figures are higher than the immediate mapping figures. It is as expected as the locations of all the fixations are known when making the deferred mapping. Both sets of figures are high however and attest to the success of both algorithms.

5.3 Assessing the Mapping of Fixations to Words

This was not done. All fixations that are mapped to words are first allocated to a line in both algorithms. The main difference then is that in deferred mapping, the horizontal locations of all of the fixations mapped to a line could be

adjusted according to the range between the left edge of the leftmost word on the line and the right edge of the rightmost word on the line. This lateral scaling cannot take place in immediate mapping. With immediate mapping, the mapping of the fixation to the word is based on the horizontal coordinate of the fixation in relation to the horizontal extents of each word. The automatic horizontal scaling in deferred mapping corresponds to the mental transformation that the human judges would attempt in estimating the likely position the reader was fixating on. The other difference is related to using effective word width in the deferred mapping, which can differ from the actual word width if the conditions described in Section 4.3 are met.

It was assumed that automatic scaling would be more accurate than the human judgment and therefore the human judges' view of the word within line being fixated would not be more accurate than that produced by the algorithm.

5.4 The Impact of Line Separation

The data quality study in Section 3 enabled the estimation of the largest number of lines of text on the screen to be five, if there was no additional mapping provided by the two algorithms. This was used as a starting point for examining the impact of line separation on mapping performance. Five separate line separations were tested in combination with the three screens containing four, five and six lines of text respectively. The largest separation corresponded to six lines of text distributed evenly over the height of the screen, and the smallest corresponded to 12 lines. The data presented in Table 2 is now re-ordered according to line separation and shown in Table 3.

Table 3: The impact of line separation on algorithm performance

Line Separation (in screen bands)	Agreement between two judges	Number of screens	IMMEDIATE mapping	DEFERRED mapping
12 (smallest)	97.9%	6	88.2%	95.5%
10.5	98.8%	11	88.9%	95.7%
9	99.1%	24	92.7%	96%
7.5	98.5%	13	93.1%	96.6%
6 (largest)	97.9%	6	93.6%	95%
	Total	60		

One would expect that the mapping performance of the algorithms would get worse as the line separation gets smaller. This appears to be the case with immediate mapping, although not with deferred mapping. This suggests that the number of lines of text on screen could be increased further before there is a significant decrease in the performance of the deferred mapping algorithm.

6 Discussion

The study has examined two aspects of using low-cost eye trackers in the context of a gaze-enabled reading aid for use in a classroom. The first is the study of quality of data that can be expected from the tracker, and the second is the performance of two algorithms intended to compensate for limitations in the accuracy of the data obtained from the eye tracker.

6.1 Data Quality Study

The study reported in this paper is not intended to provide an absolute benchmark of quality of data provided by the myGaze system nor by the X2 system of the kind advocated by Holmqvist et al (2012). Instead, it was aimed at establishing whether the low-cost myGaze eye tracker was able to provide usable data to support tracking the words read by elementary school children in a classroom environment in conjunction with fixation-to-word mapping algorithms. The X2 was included, not as a candidate tracker, but to provide a comparison with a more expensive remote eye tracker.

The criteria used to decide that the myGaze tracker would provide usable data were data loss and the offsets of fixations when viewing targets at different locations on the screen. Firstly the data loss percentages in the test immediately after calibration were 0% for both central and peripheral targets. In the tests conducted 10 minutes after initial calibration, with intermediate activity intended to simulate activity in class using a computer, these rates rose to 6.7% and 9.3% for central and peripheral targets respectively. The corresponding rates for the X2 tracker were considerably higher. This did not entail moving the laptop (and eye tracker) away from in front of the student and back again. This may well be necessary as the current desk size would not accommodate an open book and the laptop in front of the student.

The missing data rates accord fairly well with those reported by Feit et al (2017). They tested the SMI REDn Scientific tracker and the Tobii EyeX tracker. The myGaze n tracker used in this study is a low-cost version of the REDn, both trackers being produced by SMI. In Feit et al.'s study the data loss proportions for the REDn and EyeX were 2.8% and 13.1% respectively. The data loss figures for myGaze was 0% immediately after calibration and between 6.3% and 9.3% 10 minutes after calibration. The missing data rate for the X2 tracker immediately after calibration in this study (between 11.1% and 13%) accords well for the figure reported by Feit et al. for the EyeX tracker.

Considering the offset data and the myGaze n tracker, in the study immediately after calibration the interquartile range of the x -offset data lay within the bounds of the target. Similar values were obtained for the X2 tracker, bearing in mind the data losses were between 11 and 13%. The y -offsets for both trackers extended beyond the vertical bound of the target. The 75th %-ile of the X2 tracker was considerably higher than that of the myGaze n tracker. A similar picture exists in the data collected in the tests after 10 minutes. The

ranges in y have increased more than the ranges in x , and for the X2 tracker more so than for the myGaze n tracker.

The interquartile ranges can be compared with the effective target sizes reported in Feit et al.'s study. They reported that the accuracy values for the 75th %-tile across both of the trackers tested in their study were 0.58 cm in x and 0.78 cm in y . The corresponding figures for central targets in this study were 0.93 cm and 1.24 cm, and for the peripheral targets were 1.01 cm and 1.5 cm. The main differences between the studies is in the user group, the test conditions, and that the data in this study was collected 10 minutes after calibration and simulated class-based activity.

6.1.1 Limitations of the data quality study

There are several limitations of the study of data quality. First, there was no significant variation in the lighting on the desks used for the trials. The two adjacent desks used for the tests were located on the side of the classroom away from the windows. However, Feit et al. found no significant effect of differences in lighting in their study. Second, the time that the target was displayed for was possibly too long (1.8 seconds). Some participants were observed to look away from the spaceship, having decided whether or not it belonged to the country of interest, before the next target appeared. Third, the data was based on one class and one teacher, and the extent to which children moved about during the lesson could be related to a particular teaching style.

6.2 Performance of Fixation Mapping Algorithms

The approach adopted to mapping fixations to words is *heuristic*, meaning that it is based on 'rules of thumb' that seem to work, which includes some of the values used in the algorithms. It assumes that all of the text on the screen is read in one continuous session and does not accommodate interruptions to look at pictures or illustrations on the screen. Recognizing that part of the text on the screen has been read and then interrupted, and then possibly resumed, remains for future work.

The algorithms used in this study include a number of innovations in comparison with previous work. In the immediate or real-time mapping algorithm, these are: partitioning fixations into reliable and non-reliable; different modes for relative and absolute mapping; and the application of a 'reasonableness' check of the allocation of a fixation to a particular line of text through the calculation of a weighted average. While immediate mapping does track reading history, it is not based around detecting new line events.

With the deferred mapping algorithm, we have paid particular attention to *post-hoc* detecting and merging sequences of fixations that fit the left to right reading pattern that are assumed to occur when reading lines of text. This uses a process of iteratively testing and merging pairs of progressive sequences, independently of the spatial proximity to the lines of text. Only

when this process has produced a number of sequences that are the same or fewer than the number of lines of text, does the process of mapping these to the lines of text commence. WebGazeAnalyzer of Beymer and Russell (2005) uses a similar approach of clustering fixations into sequences, prior to mapping clusters onto lines of text. The lack of detail provided in the paper of how this is implemented makes it difficult to say how similar their approach is to the one we have used. There is also no evaluation reported of how successful their algorithm was in mapping fixations to lines of text.

There was also a process of fine tuning the values of the various parameters used in the algorithms. Before the evaluation study began, there was an informal testing process of both algorithms using reading data collected from a small number of elementary school students (not the participants of the main study) that enabled fine adjustments to the parameters to be made.

The performance of both mapping algorithms showed a very high success rate in mapping fixations to lines read. Where the human judges agreed completely about which fixations belonged to which lines, the deferred mapping algorithm produced mappings that agreed with them over 96% of the time. The immediate mapping algorithm produced a corresponding figure of 92.9%. These median values of agreement of fixation to line mapping per screen spanned the range of line separations. When the level of agreement is examined for each of the line separations, the deferred mapping performance is not affected by the separations used in this study. The range of levels of agreements is from 95% to 96.6%.

There is an effect of reducing line separation on the performance of the immediate mapping algorithm. With the largest line separation used (6 screen bands), this figure is 93.6%. With the smallest separation used, corresponding to 12 screen bands, this figure drops to 88.2% agreement. This latter figure accords well with the performance reported by Hyrskykari (2006b) of a 86% success rate when using immediate mapping with Verdana 11 point font and 1.5 line spacing. This corresponds to 16 screen bands in the present study, taking the screen size and resolution used by Hyrskykari into account.


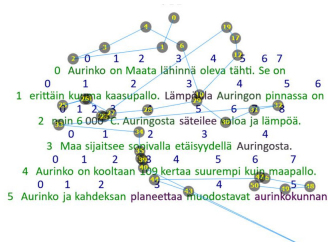
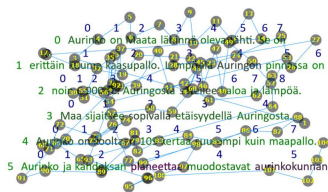

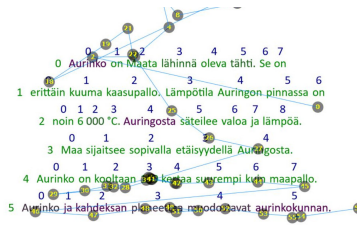
However, this does not imply that the effectiveness of the immediate mapping algorithms in both studies is similar. The level of performance in this study was obtained with children who were unconstrained using a remote low-cost tracker with low sampling rate. Hyrskykari's results were obtained in a lab using a high sampling frequency head-mounted device.

6.2.1 Reasons for excluding data from analysis of effectiveness

The success rates for both the deferred and the immediate mapping algorithms were obtained when analyzing 60 of the 72 screens for which data was collected, a loss of 17%. It is valuable to examine why the 12 screens were excluded from the analysis in order to find out whether the reasons could be predicted during classroom use.

The observations made about the cases excluded from the analysis suggest two reasons for data loss. These are presented in Table 4. First the participant

Table 4: Observations of the data collection trials and inspection of the data in those cases where data is *not* included in the analysis

Participant (Number of screens lost out of 3)	Sample of lost screen	Observations from user trial <i>Observations from data</i>
A (2 of 3)		<ul style="list-style-type: none"> Participant very focused, tries to speed up between consecutive screens, reads aloud Sat very close to the screens <i>head too close to tracker, first line tracked well then much data loss</i>
B (3 of 3)		<ul style="list-style-type: none"> Does not appear to read the text, leans on hand, appears impatient, moves a lot during the tests <i>reasonable tracking of some of top line, then appears not to read continuously, clusters of fixations on '6000C' and last word, data quality appears reasonable but not linear reading</i>
C (2 of 3)		<ul style="list-style-type: none"> Reads slowly and uses mouse to keep track of reading position, moves backwards and forwards in front of the screen <i>twice the average number of fixations for all participants and screens, noisy data but reasonable match to first 2 text lines, thereafter data too noisy to align with text lines</i>
D (3 of 3)		<ul style="list-style-type: none"> Sits still and focused during the tests, may be pretending to read <i>fixations on first line aligned well with text line but then either movement or text is not read in a linear fashion</i>
E (2 of 3)		<ul style="list-style-type: none"> Participant concentrated and reads the text <i>good alignment of fixations with the last two text lines on the screen, but much data loss on reading of first four lines</i>
Total 12		

sits in a position that makes it difficult to track the eye properly even if the text is being read (Participants A, C and E). Second the participant is not interested in the trials or is unable to read the text. It is possible that finding the text difficult to read leads to reluctance to participate (Participants B and D). Participants were told they could withdraw at any time, although none chose to do so.

These comments suggest different remedial actions. These were second grade students in an elementary school. There could be a way to provide greater motivation for the participants to read the text carefully and engage with the task. Teachers at the school suggested that gamification of reading progress would be highly appropriate. Using reading performance data over a number of sessions for an individual reader could well provide the basis for detecting and rewarding individual reading performance improvement. Hamari and Eranti (2011) suggest a framework for designing reward schemes based on game achievements. Real-time feedback about the position of the head in relation to the eye tracker could also improve tracking performance if an appropriate means of encouraging this user group to do so can be found. This kind of feedback is already used by some manufacturers. If gamification is used to motivate students to improve reading performance then keeping the head aligned with the tracker could be part of this.

6.3 Future Work

There are two directions in which work can progress from this point. One is the development of a reading aid that uses a low-cost eye tracker together the algorithms described in this paper. This aid is able to give the reader real-time support in response to what are assumed to be difficulties in reading, and enable visualization of reading performance. The durations of fixations associated with specific words can trigger, for example, automatic syllabification of that word or provide audio pronunciation. Summary presentations of difficult words can be generated for the teacher of a class, who can view the progress of a class in close to real time if each child uses their own networked reading aid. Different visualizations of reading performance have already been produced in collaboration with teachers and students in a Finnish elementary school (Špakov et al, 2017). The use of the reading aid will be investigated with teachers who are tasked with providing specialist remedial help for children with reading difficulties. Other target user groups include older students, where it is the comprehension of a word that is the issue, rather than the ability to read it. Another group is adults learning to read in a second language, possibly immigrants learning to read in the language of the home country. It is possible that the reading aid may assist in the acquisition of adult literacy skills, by providing feedback to both the learner and the teacher.

The other direction is the continued development of the processing of fixation data and the algorithms described in this paper. One objective of this would be to enable inclusion of images on the page and ensuring that the

algorithms that track reading are able to distinguish when lines of text are being read and when the reader is looking at an illustration. Beymer et al (2007) have examined this in the context of adult readers, and Hannus and Hyönä (1999) with low- and high-ability children. Additionally, if some kind of meta-activity such as gamification of reading performance could be introduced, then there is a good reason to believe that this could be exploited to improve the data quality and thereby make the performance of the algorithms more robust.

7 Conclusions

The study has shown that using a low-cost eye tracker can provide data of sufficient quality to support the construction of a gaze-enabled reading aid for use in primary schools if the system includes algorithms that map fixations to the lines of text being read. Comparable levels of performance of mapping fixations has been obtained with school children in an unconstrained use situation with a low sampling rate, low-cost tracker, that previously were obtained in a laboratory with a high sampling rate head-mounted eye tracker using adult participants.

There is a range of opportunities for a gaze-enabled reading aid that can operate in stand-alone mode or that can share reading performance data in real time. The work reported in this paper shows that it is possible to realize these opportunities, and to move the study of reading from the laboratory out into the classroom and other learning environments at a cost that makes widespread adoption of this educational technology realistic in the near future.

Acknowledgements The work was supported by the Academy of Finland as part of the GaSP project (Grant number 2501287895). We wish to thank the students of class 2C at Lamminpää School in Tampere, Finland who took part in the work reported here so enthusiastically. Our grateful thanks go as well to Matti Taimi and Suvi Taipale, members of staff at the school, for their great support. Inka Hyrskykari worked as the Research Assistant and ran the data collection trials in the classroom. We also wish to thank the reviewers of the paper for their valuable comments and input.

References

- Abdulin ER, Komogortsev OV (2015) Person verification via eye movement-driven text reading model. In: 7th International Conference on Biometrics Theory, Applications and Systems, IEEE, BTAS, DOI 10.1109/BTAS.2015.7358786, URL <http://ieeexplore.ieee.org/document/7358786/>
- Beymer D, Russell DM (2005) WebGazeAnalyzer: A system for capturing and analyzing web reading behavior using eye gaze. In: CHI '05 Extended Abstracts on Human Factors in Computing Systems, ACM, New York, NY, USA, CHI EA '05, pp 1913–1916, DOI 10.1145/1056808.1057055, URL <http://doi.acm.org/10.1145/1056808.1057055>

- Beymer D, Orton PZ, Russell DM (2007) An eye tracking study of how pictures influence online reading. In: IFIP Conference on Human-Computer Interaction, Springer, pp 456–460
- Biedert R, Hees J, Dengel A, Buscher G (2012) A robust realtime reading-skimming classifier. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, New York, NY, USA, ETRA '12, pp 123–130, DOI 10.1145/2168556.2168575, URL <http://doi.acm.org/10.1145/2168556.2168575>
- Cohen AL (2013) Software for the automatic correction of recorded eye fixation locations in reading experiments. *Behavior Research Methods* 45(3):679–683, DOI 10.3758/s13428-012-0280-3, URL <https://doi.org/10.3758/s13428-012-0280-3>
- Feit AM, Williams S, Toledo A, Paradiso A, Kulkarni H, Kane S, Morris MR (2017) Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, CHI '17, pp 1118–1130, DOI 10.1145/3025453.3025599, URL <http://doi.acm.org/10.1145/3025453.3025599>
- Hamari J, Eranti V (2011) Framework for designing and evaluating game achievements. In: DiGRA & #3911 - Proceedings of the 2011 DiGRA International Conference: Think Design Play, DiGRA/Utrecht School of the Arts, URL <http://www.digra.org/wp-content/uploads/digital-library/11307.59151.pdf>
- Hannus M, Hyönä J (1999) Utilization of illustrations during learning of science textbook passages among low- and high-ability children. *Contemporary Educational Psychology* 24(2):95–123, DOI 10.1006/ceps.1998.0987, URL <http://www.sciencedirect.com/science/article/pii/S0361476X98909870>
- Holmqvist K, Nyström M, Andersson R, Dewhurst R, Jarodzka H, van de Weijer J (2011) Eye Tracking: A comprehensive guide to methods and measures. OUP Oxford, URL <https://books.google.fi/books?id=5rIDPV1EoLUC>
- Holmqvist K, Nyström M, Mulvey F (2012) Eye tracker data quality: What it is and how to measure it. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, New York, NY, USA, ETRA '12, pp 45–52, DOI 10.1145/2168556.2168563, URL <http://doi.acm.org/10.1145/2168556.2168563>
- Hornof AJ, Halverson T (2002) Cleaning up systematic error in eye-tracking data by using required fixation locations. *Behavior Research Methods, Instruments, & Computers* 34(4):592–604, DOI 10.3758/BF03195487, URL <https://doi.org/10.3758/BF03195487>
- Hyrskykari A (2006a) Eyes in Attentive Interfaces: Experiences from Creating IDict, a Gaze-aware Reading Aid. Dissertations in Interactive Technology, University of Tampere, Department of Computer Sciences, URL <https://books.google.fi/books?id=g6cNMwAACAJ>
- Hyrskykari A (2006b) Utilizing eye movements: Overcoming inaccuracy while tracking the focus of attention during reading. *Computers in*

- Human Behavior 22(4):657 – 671, DOI <http://dx.doi.org/10.1016/j.chb.2005.12.013>, URL <http://www.sciencedirect.com/science/article/pii/S0747563205001111>, attention aware systems
- Martinez-Gomez P, Chen C, Hara T, Kano Y, Aizawa A (2012) Image registration for text-gaze alignment. In: Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces, ACM, New York, NY, USA, IUI '12, pp 257–260, DOI 10.1145/2166966.2167012, URL <http://doi.acm.org/10.1145/2166966.2167012>
- Niehorster DC, Cornelissen THW, Holmqvist K, Hooge ITC, Hessels RS (2018) What to expect from your remote eye-tracker when participants are unrestrained. Behavior Research Methods 50(1):213–227, DOI 10.3758/s13428-017-0863-0, URL <https://doi.org/10.3758/s13428-017-0863-0>
- Palmer C, Sharif B (2016) Towards automating fixation correction for source code. In: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ACM, New York, NY, USA, ETRA '16, pp 65–68, DOI 10.1145/2857491.2857544, URL <http://doi.acm.org/10.1145/2857491.2857544>
- Rayner K (1998) Eye movements in reading and information processing: 20 years of research. Psychological Bulletin 124(3):372–422, DOI 10.1037/0033-2909.124.3.372
- Reingold EM (2014) Eye tracking research and technology: Towards objective measurement of data quality. Visual Cognition 22(3-4):635–652, DOI 10.1080/13506285.2013.876481
- Sanches CL, Kise K, Augereau O (2015) Eye gaze and text line matching for reading analysis. In: Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, ACM, New York, NY, USA, UbiComp/ISWC'15 Adjunct, pp 1227–1233, DOI 10.1145/2800835.2807936, URL <http://doi.acm.org/10.1145/2800835.2807936>
- Sharmin S, Špakov O, Rähä KJ (2013) Reading on-screen text with gaze-based auto-scrolling. In: Proceedings of the 2013 Conference on Eye Tracking South Africa, ACM, New York, NY, USA, ETSA '13, pp 24–31, DOI 10.1145/2509315.2509319, URL <http://doi.acm.org/10.1145/2509315.2509319>
- Sibert JL, Gokturk M, Lavine RA (2000) The reading assistant: Eye gaze triggered auditory prompting for reading remediation. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology, ACM, New York, NY, USA, UIST '00, pp 101–107, DOI 10.1145/354401.354418, URL <http://doi.acm.org/10.1145/354401.354418>
- Stampe DM, Reingold EM (1995) Selection by looking: A novel computer interface and its application to psychological research. Studies in Visual Information Processing 6:467–478
- Špakov O, Siirtola H, Istance H, Rähä KJ (2017) Visualizing the reading activity of people learning to read. Journal of Eye Movement Research 10(5), DOI 10.16910/jemr.10.5.5

Špakov O, Istance H, Viitanen T, Siirtola H, Rähä KJ (2018) Enabling unsupervised eye tracker calibration by school children through games. In: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ACM, New York, NY, USA, ETRA '18, pp 36:1–36:9, DOI 10.1145/3204493.3204534, URL <http://doi.acm.org/10.1145/3204493.3204534>

A Flowcharts: Fixation mapping deferred until all fixations have been recorded (section)

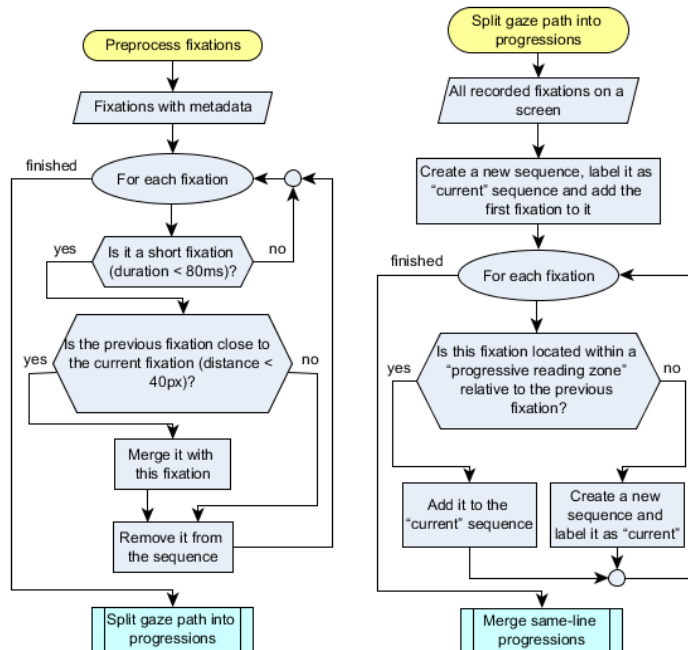


Fig. 14: deferred mapping preparation
(left) merging short closely situated fixations, (right) splitting all recorded fixations into progressive sequences

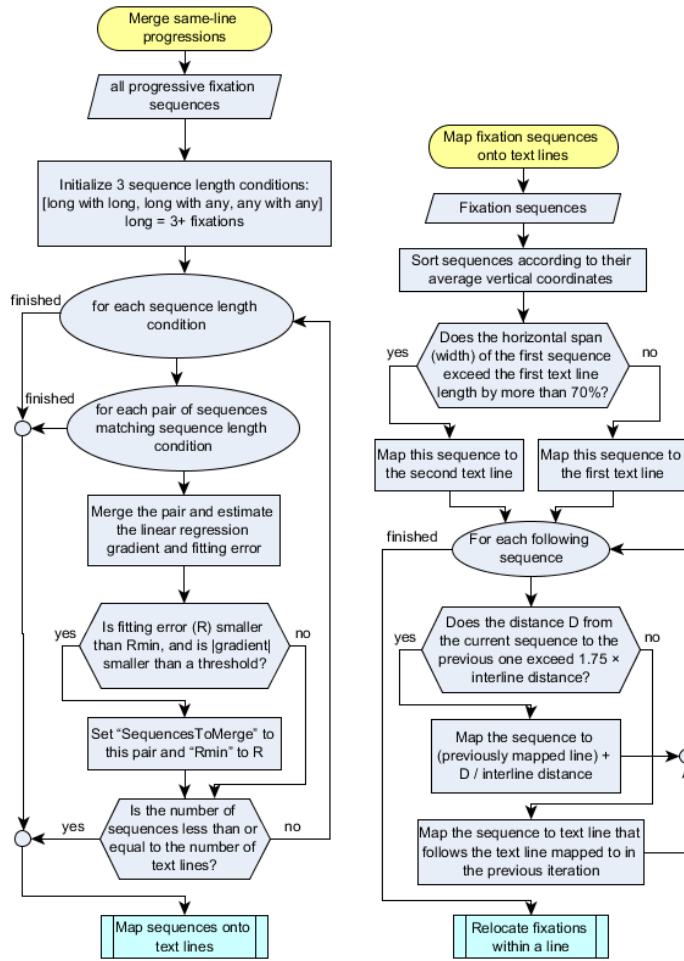


Fig. 15: Producing fixation sequences corresponding to lines of text (left) merging sequences that belong to the same line of text, (right) mapping fixation sequences to lines of text

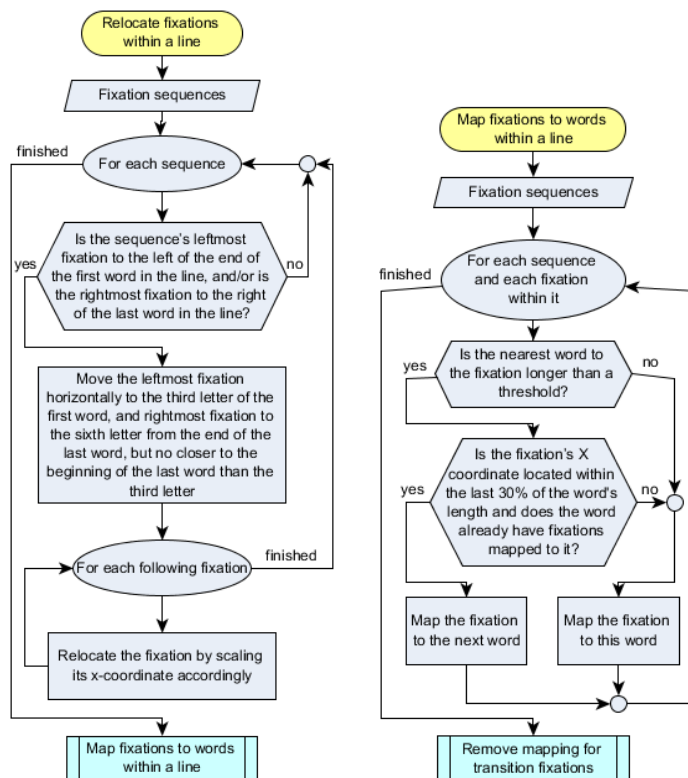


Fig. 16: Mapping fixations within a sequence to words within a line of text (left) adjusting the horizontal position of fixations, (right) mapping fixations to words

B Pseudocode: Mapping current fixation relative to previously mapped fixations (section 4.2.1)

Fixations:

fixation = fixation being mapped
 lrmF = last reliable mapped fixation

Text Lines:

pml = provisionally mapped line
 Lpfmt = line previous fixation mapped to

```
let vertical separation = fixation.y - lrmF.y;

if (fixation.y is equal or below lrmF.y) {
  if (vertical separation < fontsize * 1.2) {
    provisionally map fixation to Lpfmt;
    fixation mapping will be reliable if mapping confirmed;
    CheckProvisionalMapping( fixation )
  }
  else if (vertical separation < interline separation * 1.75) {
    provisionally map fixation to Lpfmt + 1;
    fixation will NOT be reliable if mapping confirmed;
    CheckProvisionalMapping( fixation )
  }
  else
    AbsoluteMap( fixation )
}
else {
  provisionally map fixation to (Lpfmt - round( vertical separation /
    interline separation ));
  CheckProvisionalMapping( fixation )
}
map fixation to word within the line
```

B.1 CheckProvisionalMapping (section 4.2.2)

```
for all lines with previously mapped fixations {
  calculate average vertical distance of fixation mapped to it to
    line.center.y;
  calculate weight to give this average value based on inverse of
    distance line.center.y to fixation.y;
}
let w = sum of weighted average vertical distances

if (abs( fixation.y - pml.y ) <= w * interline separation * 0.5) {
  confirm provisional mapping of fixation to pml;
  flag fixation as being Reliable or Not Reliable;
}
else {
  /* reject provisional mapping and use absolute mapping */
  AbsoluteMap( fixation );
}
```


C Pseudocode: Absolute mapping with no reference to previously mapped fixations (section 4.2.3)

Fixations:

fixation = fixation being mapped

pmrF = previously mapped reliable fixations

Text Lines:

Lines = list of horizontal text rectangular lines with a center and extends

```

/* create List of candidate Lines to map fixation to */
let nearest lines list = an empty list of candidate lines;

for each line in lines of text on current page {
  yoffset = fixation.y - line.center.y;
  if (fixation.x < line.extent.left)
    xoffset = line.extent.left - fixation.x;
  else if (fixation.x > line.extent.right)
    xoffset = fixation.x - line.extent.right;
  else
    xoffset = 0;

  if (abs( yoffset ) < interline separation * 0.7) AND
    (xoffset < 100px) {
    add this line to the nearest lines list;
  }
}

/* list now contains 0, 1 or 2 lines */
If (nearest lines list.length is 0)
  fixation is not mapped at all;
else if (nearest lines list.length is 1)
  map fixation to that line;
else { /* nearest lines list.length is 2 */
  for each line in nearest lines list {
    normalised yoffset = yoffset / interline separation * 0.7;
    normalised xoffset = xoffset / 100px;
    inverse proximity score (IPS) = normalised yoffset +
      normalised xoffset;
  }

  Difference in inverse proximity scores (DIPS) = first line IPS -
    second line IPS;

  /* map fixation to first or second line according to DIPS */
  DIPS threshold = 0;
  if (first line has no pmrF) OR (second line has no pmrF) {
    adjust DIPS threshold by +/- 0.3 to favor selecting the line
    without pmrF;
  }

  if DIPS < DIPS threshold
    map fixation to first line;
  else
    map fixation to second line;
}

```

D Pseudocode: Fixation mapping deferred until all fixations have been recorded (section 4.3)

D.1 Preprocessing and building sequences of fixations

```

/* pre-processing list of recorded fixations */
for each fixation in a list of fixations recorded on a page of text {
    if (fixation.duration < threshold1)
        if (fixation distance to previous fixation < threshold2)
            merge fixation with previous fixation;
        else
            remove fixation from list;
    else if (fixation distance to nearest word > threshold3)
        remove fixation from list;
}

/* build progressive horizontal sequences of fixations */
let sequences = a list of progressive sequences of fixations
let sequence = new Sequence added to sequences;
add first fixation to sequence;

for each fixation in remaining fixations {
    if (fixation is recognized as left-to-right reading fixation)
        add fixation to sequence;
    else {
        sequence = new Sequence added to sequences;
        add fixation to sequence;
    }
}

```

D.2 Merge sequences of fixations

```

/* phase 1: constraints on both current Sequence and candidate
sequence for merger */
repeat
    for each sequence1 of 3 or more fixations {
        calculate gradient and regression error for sequence;
        for each other sequence2 of 3 or more fixations {
            form combined sequence of sequence1 and sequence2;
            calculate gradient and regression error of combined
            sequence;
        }
        select combined sequence with minimum regression error and
        with both gradient and regression error lying within
        acceptable threshold value;

        if (sequence combination exists)
            remove sequence1 and sequence2 from sequences;
            add combined sequence to sequences;
    }
until no new mergers result

/* phase 2: constraints on candidate sequence for merger, but not on
current Sequence */
repeat
    for all sequences of any number of fixations

```

```

    /* as phase 1 */
    for all other sequences of 3 or more fixations
        /* as phase 1 */
    /* as phase 1 */
until no new mergers result

/* phase 3: no constraints in sequence length */
repeat
    for all sequences of any number of fixations{
        /* as phase 1 */
        for all other sequences of any number of fixations
            /* as phase 1 */
        /* as phase 1 */
    }
until no new mergers result

/* optional phase */
if (number of sequences > number of text lines)
    repeat
        /* same as phase 3 but without gradient restriction */
    until number of sequences = number of text lines

```

D.3 Mapping sequences of fixations to lines of text

```

for each sequence in sequences
    calculate average Y of sequence;

sort sequences by average Y;

if (first sequence width is NOT much longer than first line width)
    map first (topmost) sequence to first line of text;
else
    /* first line maybe a short title */
    map first (topmost) sequence to second line of text;

/* map remaining sequences to remaining lines in order */
for each sequence in remaining sequences {
    map sequence to next unmapped line;
    let difference = sequence.y - previousSequence.y;
    if (difference > 1.7 * average line separation)
        advance the line the fixations are mapped to by (difference /
            line-separation - 1);
}

```

D.4 Mapping fixations in a sequence to word in a line of text

```

/* Relocating fixations in a sequence within the line it has been
   mapped to */
for each sequence in sequences {
    if (sequence.firstFixation.x is left of sequence.line.extent.left)
        consider moving this to 3rd letter of first word in the line;

    if (sequence.lastFixation.x is right of sequence.line.extent.right
        )
        consider moving this to 6th letter from end of last word in
        the line;

    if (sequence.firstFixation or sequence.lastFixation were moved) {

```

```
        restrict the change in sequence span by up to a maximum of 10%
            of original sequence span;
    for each other fixations in sequence {
        move and scale fixation.x accordingly;
    }
}

/* Mapping fixations to words within lines */
for each sequence in sequences {
    for each fixation in sequence {
        identify nearest word in the line to fixation.x;
        if (nearest word.width > threshold4) OR (previous fixations in
            sequence have been mapped to this word)
            nearest word.effectiveWidth = nearest word.width * 0.7;
        else
            nearest word.effectiveWidth = nearest word.width;

        if (fixation.x is between [nearest word.extent.left AND
            nearest word.extent.left + nearest word.effectiveWidth])
            map fixation to nearest word;
        else
            map fixation to next closest word;
    }
}
```